A Simple yet Accurate Autoadaptive Model of Network Traffic for Detection of Attacks on Low Latency Services

Rémi Cogranne, Marius Letourneau, Guillaume Doyen, Huu Nghia Nguyen

A Simple yet Accurate Autoadaptive Model of Network Traffic for Detection of Attacks on Low Latency Services

Rémi Cogranne*
Troyes University of Technology
Troyes, France
remi.cogranne@utt.fr

Marius Letourneau
Troyes University of Technology
Troyes, France
marius.letourneau@utt.fr

Guillaume Doyen
OCIF - IRISA (UMR CNRS 6074)
IMT Atlantique,
Rennes, France
guillaume.doyen@imt-atlantique.fr

Huu Nghia Nguyen†
Montimage
Paris, France
huunghia.nguyen@montimage.eu

ABSTRACT

This paper addresses the problem of detection of attacks in computer networks. More precisely, we consider attacks on emerging low-latency services, which typically require a specific traffic management system. We present a simple yet very efficient hybrid method that takes advantage of both autoencoders and transformer models. The original method is compared with the current state-of-the-art on a large real-life dataset of network traffic to show the relevance of the proposed approach, especially for low false-positive rates. A quick ablation analysis shows that the efficiency of the method relies on the combined use of the two approaches jointly in our hybrid model.

CCS CONCEPTS

• **Theory of computation** → **Unsupervised learning and clustering**; • **Networks** → **Network security**; *Denial-of-service attacks*; • **Computing methodologies** → **Artificial intelligence**; *Simulation evaluation*; Unsupervised learning; • **Computer systems organization** → *Real-time system architecture*; • **General and reference** → Experimentation; • **Applied computing** → Network forensics.

KEYWORDS

Multivariate Time-Series, Network Security, Multidimensional Signal Processing, Attack detection, Cybersecurity

ACM Reference Format:
Rémi Cogranne, Marius Letourneau, Guillaume Doyen, and Huu Nghia Nguyen. 2025. A Simple yet Accurate Autoadaptive Model of Network Traffic for Detection of Attacks on Low Latency Services. In *Proceedings of 10th ACM International Conference on Multimedia Systems and Signal Processing (ICMSSP 2025) (ACM ICMSSP'25).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding author.
†This work was partially supported by the French ANR MOSAICO project No. ANR-19-CE25-0012.

ACM ICMSSP'25, May 9 – 11, 2025, Fukui, Japan

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The technological progression of network infrastructures such as the deployment of fibre-optic links, fifth-generation mobile networks (5G), Virtualized Network Functions (VNFs), and network slicing techniques, to cite a few, have opened an age of significantly elevated throughput rates and slashed latency periods [1, 10]. These advancements have given rise to a variety of new applications with intrinsically low latency demands, such as cloud gaming, cloud robotics, tele-robotics, and tactile internet services. However, the development of novel Internet services opens new possible vectors of attacks and security must be a priority because, without any doubt, in today's digital age, the likelihood of an attack increases along with the potential impact, and low latency services will become a target when deployed. In addition, as often seen in computer networks, Future Internet will not replace existing services, hence the birth of a coexistence of regular internet traffic, some with high-throughput targets, with such emerging latency-sensitive applications.

The Low Latency, Low Loss, and Scalable Throughput (L4S) architecture has been positioned as a prospective solution to meet these cutting-edge network demands. When tested, the L4S architecture demonstrated satisfying performance under standard traffic behaviour [19]. Yet, research lacks insight regarding its capability to work under abnormal or non-regulated traffic conditions. For instance, it has been shown in [4] how rather moderate traffic bursts can heavily impact L4S performance, thus preventing the delivery of intended Quality of Experience (QoE). This problem is even more concerning in the context of cyberattacks; for instance, it has been shown in [14] how it is possible to compromise cloud gaming platform services by leveraging so-called booters making a game become very unfair and ultimately compromising players' experience.

The present paper addresses the problem of the detection of attacks that specifically aims at disrupting low-latency services. As for a vast majority of similar DDoS detection problems, there are several major intrinsic difficulties. First the attacks on low-latency

services are rather new; their means are not completely known and entirely characterized. Second, the traffic of a computer network is, in general, a complex signal that is difficult to analyse and model. Last and not least, the detection methodology must be reliable in the sense that it should be able to detect (1) attack with a limited delay, as quickly as possible and (2) it should prevent as much as possible false-alarm rate in order to avoid raising unwanted false alarms, which would undermine the overall system's usability.

We propose an original methodology based on an autoadaptive linear model for analysing network traffic measurements, considered in this paper as multivariate time series. Our model leverages the advantages of detection theory using a linear model, which is well established, and the relevance of unsupervised learning to build automatically an accurate model representing the multivariate time series under inspection.
In our prior works [16, 17], we have described the impact of a specific attack so-called "unresponsive ECN" and assessed its practical efficiency throughout large-scale numerical evaluation and experimentation. Therefore, the detection method proposed in the present paper is evaluated on a large dataset of real network traffic data implemented in a realistic testbed with several legitimate clients and attackers.
We show that the proposed method, despite its simplicity, performs very well for multidimensional signal processing and detection.

The present paper is organized as follows. Section 2 provides a brief overview of low-latency network architectures currently being transferred from research labs to the real world. We recall the possible attack mechanisms on low-latency services and their impact on the traffic. Section 3 states the problem of dealing with nuisance parameters and presents the methodology used in the present work to represent and reject the legitimate traffic. Then Section 4 presents the autoadaptive linear model used to represent the multidimensional data that computer traffic networks are. It especially explains how unsupervised learning is used to adjust the proposed linear model with the previous observations in order to represent the traffic metrics accurately. Section 5 presents the proposed method for the detection of low-latency attacks. Section 6 assesses the proposed methodology on a large real-world dataset of low-latency attacks and legitimate traffic. Section 7 concludes the paper and draws possible future works.

2 SUMMARY OF ATTACKS ON LOW-LATENCY SERVICES

In a nutshell, the L4S architecture rests on three main simple pillars. On the client endpoint side, L4S leverages accurate Explicit Congestion Notifications (ECN), which provides real-time indicators of network congestion to alert users such that they shall limit their traffic before a large congestion occurs. It also implements a novel congestion control algorithm (CCA), namely Prague, available both for the Transmission Control Protocol (TCP) and the Quick UDP Internet Connections (QUIC) protocol, to use as much as possible link bandwidth while avoiding traffic congestion [20, 25]. On the network side, L4S uses a Dual-Queue Coupled Active Queue Management (AQM) system which has the ability to handle queueing priorities, coupled directly with a unique packet classifier that segregates signals for Low Latency (LL) and regular, legacy flows. This categorical discrimination is instrumental in a bid to institute targeted behaviour modification, allowing LL flows not to disproportionately influence the Classic bandwidth. The central performance of the dual AQM relies on this coupling mechanism, which shall find an equilibrium in moderating the classical queue, for legacy traffic, in order to preserve low-latency features for the second queue while preventing the low-latency flows' tendency to cause a so-called starvation of classic, legacy traffic [1, 11].

In our previous work [17], we demonstrated that traffic that does not comply with the rules of ECN and TCP Prague protocols can significantly degrade the effective operation of L4S. Building on this, the present paper focuses on a specific type of malicious flow: "unresponsive yet ECN-compatible" flows. These flows pretend to adhere to ECN congestion notifications but fail to adjust their sending rates accordingly, thereby subverting the intended behaviour. It is worth noting that a small proportion of unresponsive traffic is inherent in computer network traffic because of UDP and VoIP communications, for instance. Fortunately, L4S is designed to accommodate a limited fraction of such flows, as acknowledged in [27]. However, as the proportion of unresponsive flows increases, L4S struggles to maintain both low-latency services and the throughput of regular flows. Alarmingly, even a tiny fraction of "unresponsive yet ECN-compatible" flows can rapidly degrade L4S performance and undermine any low-latency services.

We have also conducted large-scale experimentation in [16] to characterize the attack mechanism on L4S. Considering the metrics that a router can easily monitor, we showed that the impact of the attack is entirely characterized by the following seven measurements:

(1) traffic rate, or number of bytes transmitted per second ;
(2) in-router queueing delay, which is the average time a packer remains in the cache of the router before being retransmitted (for both low-latency and regular traffic) ;
(3) the number of packets within both queues, which is the fraction of the total number of packets in the router's cache memory ;
(4) the probability of ECN marking, which is the probability used by the dual queue system of setting ECN notification ;
(5) probability of a packet drop, a packet is dropped when either the router drops it preventively or when its queue is fully occupied ;
(6) the total number of ECN marks written on packets by the AQM, which can be due to natural queue-building behaviours within each queue that lead to an increase in the marking probability ;
(7) the number of ECN marks that is due to a step threshold overflow of the low latency queue ;

It is especially interesting to note that a Principal Component Analysis shows that some of those metrics are extremely correlated under normal conditions ; yet it is shown in [16] that when a large fraction of unresponsive flows occur, the correlation between these metrics changes significantly and is much less obvious in general. Therefore, in the present paper, we have decided to keep the very same set of metrics altogether even though they are not always linearly independent.

3 DEALING WITH NUISANCE PARAMETERS

Generally speaking, the problem we address in the paper is the detection of a signal (the impact of unresponsive traffic) in the presence of so-called nuisance parameters. Indeed, legitimate traffic constitutes a complex "non-anomalous" environment: the legitimate traffic is not constant: it can change depending on the users' behaviours. The legitimate traffic is generally unknown and it is not straightforward to model. In the present paper, we adopt a general statistical model: the legitimate traffic is assumed to be drawn from the following Gaussian distribution:

$$\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t), \quad \text{under legitimate traffic,} \tag{1}$$

where $\mathcal{N}(\cdot)$ represents the Gaussian, or normal, distribution, $\mathbf{x}_t$ represents the metrics at time t whose expected value is denoted $\boldsymbol{\mu}_t$ and with covariance matrix Σ_t.
On the opposite, when an attack on low-latency services is launched, the traffic is modelled by a sum of the legitimate traffic and the impact of the attack:

$$\mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t + \mathbf{a}_t, \Sigma_t), \quad \text{under attack.} \tag{2}$$

In other words, it is assumed that the attack impacts the expectation of the observation and not much the covariance matrix, which remains almost unchanged. In addition, the anomaly $\mathbf{a}_t$ is assumed constant ; more precisely, the impact of the attack can vary in intensity but its "footprint" remains the same, hence $\mathbf{a}_t = \eta_t \mathbf{a}$ with $\mathbf{a}$ the constant "footprint" and η model the "intensity" of the attack.

Usually, the first step when dealing with correlated observations is to apply the so-called whitening transformation with the matrix $\Sigma_t^{-1/2}$ satisfying ${\Sigma_t^{-1/2}}^\top \Sigma_t^{-1/2} = \Sigma_t$. Usually, $\Sigma_t^{-1/2}$ is obtained as the Chelesky decomposition of the covariance matrix Σ_t. Indeed, it is straightforward from (1)–(2) that:

$$\mathbf{x}'_t = {\Sigma_t^{-1/2}}^\top \mathbf{x}'_t \sim \mathcal{N}(\boldsymbol{\mu}'_t, \mathbf{I}), \quad \text{under legitimate traffic,} \tag{3}$$

With $\boldsymbol{\mu}'_t = {\Sigma_t^{-1/2}}^\top \boldsymbol{\mu}_t$.

It is obvious from Equations (1)–(3) that the legitimate traffic $\boldsymbol{\mu}_t$ represents a nuisance parameter in the sense that it has no interest for the detection of the signal of interest, namely the attack footprint $\mathbf{a}$, but it shall be carefully taken into account, as it may obfuscate the attack it is aimed at detecting.
In the present paper, we adopt a linear model of the legitimate traffic:

$$\boldsymbol{\mu}_t = \mathbf{H}_t \theta_t, \tag{4}$$

where the column of the matrix $\mathbf{H}_t$ spans the subspace of the legitimate traffic at time t.

The main advantage of the linear model (4) is that it easily allows removing the nuisance parameters $\boldsymbol{\mu}_t = \mathbf{H}_t \theta_t$. Indeed, the rejection of the nuisance parameter can be carried out by simply projecting the vector of network traffic measurements $\mathbf{x}_t$ onto the orthogonal complement $R(\mathbf{H}_t)^\perp$ of the legitimate traffic, which is the subspace spanned by the column matrix $\mathbf{H}_t$. This projection can be defined by the projector of the Generalized Least-Square (GLS) as follows:

$$\mathbf{P}^\perp_{\mathbf{H}_t} = \mathbf{I}_p - \mathbf{H}_t \left(\mathbf{H}_t^\top \Sigma_t^{-1} \mathbf{H}_t \right)^{-1} \mathbf{H}_t^\top \Sigma_t^{-1}. \tag{5}$$

Obviously, the second term of matrix $\mathbf{P}^\perp_{\mathbf{H}_t}$ corresponds to the least square estimation of the nuisance parameter $\mathbf{H}_t \theta_t$ under the assumption that the covariance matrix Σ_t is proportional to the identity matrix. Hence, the projector $\mathbf{P}^\perp_{\mathbf{H}_t}$ corresponds to the subtraction of this estimation of legitimate traffic from the observation.
Alternatively, the rejection of the nuisance parameter can be achieved with the orthonormal matrix $\mathbf{W}_t = (w_1, \ldots, wp - q)$, where w_i are the eigenvectors of the projection matrix $\mathbf{P}^\perp_{\mathbf{H}_t}$ corresponding to eigenvalues 1.
The matrix $\mathbf{W}$ verifies the following properties:

$$\mathbf{W}\mathbf{H} = \mathbf{0}\,, \quad \mathbf{W}^\top \mathbf{W} = \mathbf{P}\mathbf{H}^\perp\,, \quad \mathbf{W}\mathbf{W}^\top = \mathbf{I}p - q. \tag{6}$$

The rejection of a linear nuisance parameter can be simply carried out as

$$\mathbf{W}_t^\top \mathbf{x}_t. \tag{7}$$

It immediately follows from Equations (1), (2) and (7) that:

$$\mathbf{W}_t^\top \mathbf{x}'_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}p - q), \quad \text{under legitimate traffic,} \tag{8}$$

and:

$$\mathbf{W}_t^\top \mathbf{x}'_t \sim \mathcal{N}(\mathbf{a}^\perp, \mathbf{I}p - q), \quad \text{under attack,} \tag{9}$$

where $\mathbf{a}^\perp = \mathbf{W}_t^\top \mathbf{a}$ represents the impact of the attack on low-latency services it is aimed at detecting.

4 PROPOSED AUTOADAPTIVE LINEAR MODEL FOR REPRESENTING NETWORK TRAFFIC

The model presented in the previous section 3 is very general in statistical decision theory [12, 13, 26] and has been widely used in image processing [6–9, 28, 29] and computer network traffic modelling [2, 3, 18, 22, 23, 31]. However, it is not accurate enough for the targeted application of detection of attacks on low-latency services to represent the network traffic used in the present work. To design a model that is accurate enough to represent the legitimate network traffic and, yet, to preserve the simplicity of the linearity, we propose an original autoadaptive model [5, 24, 30, 31]. The principle of this model is depicted in Figure 1. First the measurements are gathered and the set of L last measurements $\mathbf{X}_t = \mathbf{x}_{t-L+1}, \mathbf{x}_{t-L+2}, \ldots, \mathbf{x}_t$ are analysed jointly. These samples are modelled altogether using a linear parametric model:

$$\mathbf{X}_t \sim \mathcal{N}(\mathbf{H}_t \theta_t, \Sigma_t), \quad \text{under legitimate traffic,} \tag{10}$$

The $M = kL$ previous samples are used to (1) estimate the covariance matrix Σ_t and to build the linear model $\mathbf{H}_t$ adaptively. To this end we perform a principal component analysis (PCA) on the k windows of size L using the $M = kL$ previous observations $x_{t-2L-M+1}, \ldots, x_{t-2L}$,. Let us denote $\mathbf{p}_1, \ldots, \mathbf{p}_L$ the principal components sorted by the associated eigenvalues. The linear model is built with the n principal components $\mathbf{H}_t = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$.
Note in our case, we tested several configurations, and we kept the one that gave the best results: $L = 8$ for a total of $p = 64$ metrics analysed jointly with a number of components $n = 3$.
Also, note that, as depicted in Figure 1 some between the "detection window" $\mathbf{x}_{t-L+1}, \ldots, \mathbf{x}_t$ and the "parameters estimation window" $x_{t-2L-M+1}, \ldots, x_{t-2L}$, are not used. Indeed, the idea of the proposed autoadaptive linear model we proposed is to carry out the estimation and the rejection with observations under legitimate

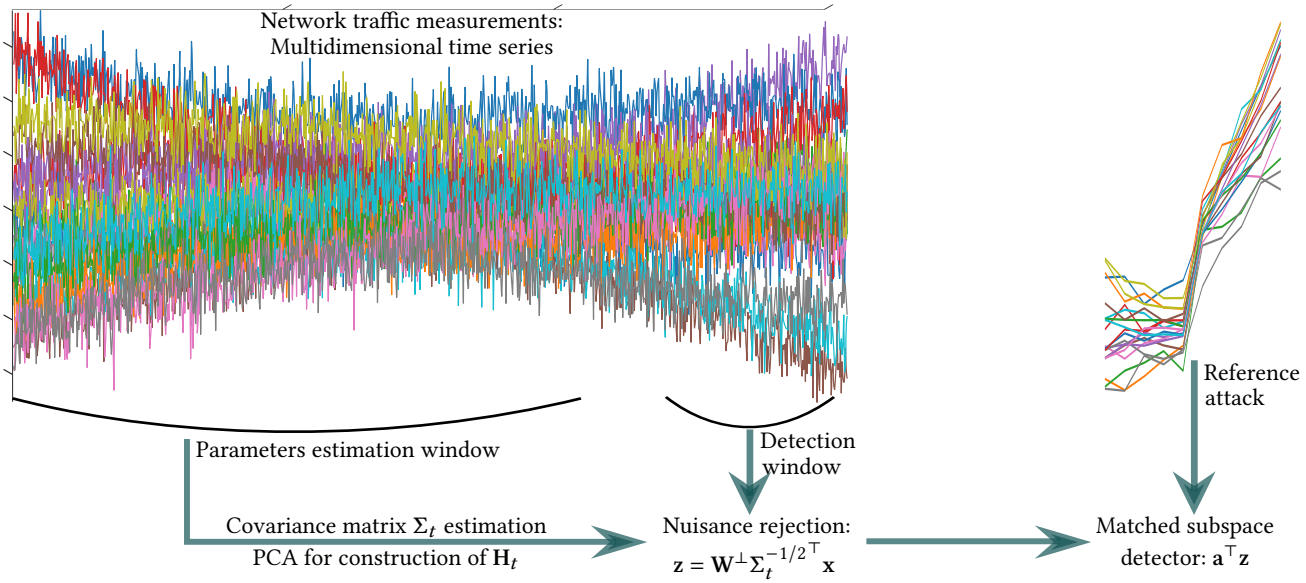

Figure 1: Illustration of the architecture of the proposed model. The network traffic measurements, represented as a multivariate time series, are firstly input into the autoencoder, whose goal is to learn the underlying model and representation of legitimate traffic. Then the residuals are considered by subtracting the output of this autoencoder and the result is fed to a multi-headed self-attention module before classification.

traffic, in other words, that are free from the impact of a potential attack on low-latency services. A certain detection delay may be needed to ensure that the attack is detected; hence the need for a certain distance between the parameter estimation window and the observation used for the detection.

With this idea of using samples without the impact of a potential attack, the last originality of the proposed autoadaptive linear model is to fix the parameter estimation window when an attack is detected. While the two windows are "sliding" with the observation under legitimate traffic, we keep on using the same samples for parameter estimation as long as an attack is detected under the "detection window" to ensure that the estimation is, as much as possible, free from the impact of a potential attack.

Last but not least, note that we assume that noise corrupting the metrics is not correlated in time; hence the statistical independence between x_{t-1} and x_t. This allows us to reduce the estimation of the covariance matrix Σ_t by the covariance between metrics.

5 METHODOLOGY FOR DETECTION OF ATTACKS ON LOW-LATENCY SERVICES

Using the original autoadaptive model described in section 4 we can eventually describe the observed metrics of the computer networks as:

$$\begin{cases} \text{under } \mathcal{H}_0: & \mathbf{X}_t \sim \mathcal{N}(\mathbf{H}_t\theta_t, \Sigma_t), \\ \text{under } \mathcal{H}_1: & \mathbf{X}_t \sim \mathcal{N}(\mathbf{H}_t\theta_t + \mathbf{a}, \Sigma_t). \end{cases} \tag{11}$$

As explained in Section 3 after whitening transformation and rejection of the legitimate traffic $\mathbf{H}_t\theta_t$ the statistical detection problem(11) can be rewritten, under the principle of invariance, see [15][Chap. 4], as:

$$\begin{cases} \text{under } \mathcal{H}_0: & \mathbf{W}_t^\top\mathbf{X}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \text{under } \mathcal{H}_1: & \mathbf{W}_t^\top\mathbf{X}_t \sim \mathcal{N}(\mathbf{a}^\perp, \mathbf{0}). \end{cases} \tag{12}$$

where $\mathbf{a}^\perp = \mathbf{W}_t^\top\mathbf{a}$ represents the impact of the attack on low-latency services it is aimed at detecting.

The statistical detection problem (12) is a test between two simple hypotheses. According to the Neyman-Pearson Lemma [15, Theorem 3.2.1], the optimal, most powerful test is the likelihood ratio test. Because the statistical detection problem (12) can be reduced to detecting a known signal $\mathbf{a}^\perp$ in Gaussian noise, we can calculate the log-likelihood ratio between the two hypotheses (12) for an observation $\mathbf{X}_t$ as:

$$\Lambda(\mathbf{X}_t) = \log \frac{(2\sqrt{\pi}\sigma)^{-(p-q)} \exp\left(-(2\sigma^2)^{-1}\|\mathbf{W}_t^\top\mathbf{X}_t - \mathbf{W}_t^\top\mathbf{a}\|^2\right)}{(2\sqrt{\pi}\sigma)^{-(p-q)} \exp\left(-(2\sigma^2)^{-1}\|\mathbf{W}_t^\top\mathbf{X}_t\|^2\right)} \tag{13}$$

$$= -\|\mathbf{W}_t^\top\mathbf{X}_t - \mathbf{W}_t^\top\mathbf{a}\|^2 + \|\mathbf{W}_t^\top\mathbf{X}_t\|^2 \tag{14}$$

where $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top\mathbf{x}$ is the Euclidean norm, $p = 64$ is the dimension of $\mathbf{X}_t$ and $n = 3$ is the number of principal components used to design the matrix $\mathbf{H}_t$.

Obviously, the log-likelihood ratio established in the Equation (14) depends on the observation $\mathbf{X}_t$ only through the term:

$$\Lambda^\star(\mathbf{X}_t) = \left(\mathbf{W}_t^\top\mathbf{X}_t\right)^\top \mathbf{W}_t^\top\mathbf{a} = \mathbf{X}_t^\top\mathbf{W}_t\mathbf{W}_t^\top\mathbf{a} = \mathbf{X}_t^\top\mathbf{P}_{\mathbf{H}}^\perp\mathbf{a}, \tag{15}$$

which correspond to the well-known matched subspace detectors [**?**] after rejection of the non-anomalous background via $\mathbf{W}_t^\top$.

In practice the covariance matrix $\Sigma_t^{-1/2^\top}$ and the linear model $\mathbf{H}_t$ are estimated from the previous observations, as explained in the section 4.

Similarly, since the network traffic is widely varying the decision statistics $\Lambda^\star(\mathbf{X}_t)$ do not always have a zero mean under legitimate traffic. Therefore, the "parameter estimation window" is used to estimate the mean of the decision statistic $\Lambda_0^\star$ which is subtracted from the LR calculated over the detection window:

$$\Lambda^\star(\mathbf{X}_t) = \mathbf{X}_t^\top \mathbf{P}_{\mathbf{H}}^\perp \mathbf{a} - \Lambda_0^\star. \tag{16}$$

The statistical test is finally defined formally by:

$$\delta(\mathbf{X}_t) = \begin{cases} \mathcal{H}_0 \text{ if } \Lambda^\star(\mathbf{X}_t) = \mathbf{X}_t^\top \mathbf{P}_{\mathbf{H}}^\perp \mathbf{a} - \Lambda_0^\star < \tau \\ \mathcal{H}_1 \text{ if } \Lambda^\star(\mathbf{X}_t) = \mathbf{X}_t^\top \mathbf{P}_{\mathbf{H}}^\perp \mathbf{a} - \Lambda_0^\star \geq \tau \end{cases} \tag{17}$$

where τ is the decision threshold.

The detection methodology is summarized in algorithm 1 which clearly explains the role of the estimation window, the detection window and how the statistical test (14)–(16) is calculated using real network traffic metrics.

Algorithm 1 Low-lantency attack detection using the autoadaptive model

Require: L ▷ Detection window size
Require: $M = kL$ ▷ Estimation window size
Require: $\mathbf{X}_t = (x_{t-L+1}, \ldots, x_t)$ ▷ Detection window
Require: $\mathbf{X}_0 = (x_{n-M+1}, \ldots, x_n) = (\mathbf{X}_1, \ldots, \mathbf{X}_k)$ ▷ Estimation window
Using the estimation window $\mathbf{X}_0$:
$\mathbf{H}_t \leftarrow PCA(\mathbf{X}_1, \ldots, \mathbf{X}_k)$
$\Sigma_t \leftarrow Covariance(\mathbf{X}_0)$
$\mathbf{P}_{\mathbf{H}_t}^\perp \leftarrow \mathbf{I}_p - \mathbf{H}_t \left(\mathbf{H}_t^\top \Sigma_t^{-1} \mathbf{H}_t\right)^{-1} \mathbf{H}_t^\top \Sigma_t^{-1}$
$\Lambda_0^\star \leftarrow Mean(\mathbf{X}_1^\perp \mathbf{P}_{\mathbf{H}_t}^\perp \mathbf{a}, \ldots, \mathbf{X}_k^\perp \mathbf{P}_{\mathbf{H}_t}^\perp \mathbf{a})$
Using the detection window $\mathbf{X}_t$:
$\Lambda^\star(\mathbf{X}_t) \leftarrow \mathbf{X}_t^\perp \mathbf{P}_{\mathbf{H}_t}^\perp \mathbf{a} - \Lambda_0^\star$
if $\Lambda^\star(\mathbf{X}_t) \leq \tau$ **then**
  $n = t - M - L$ ▷ The estimation window is moved to the previous metrics
else if $\Lambda^\star(\mathbf{X}_t) > \tau$ **then**
  $\mathbf{X}_0 \leftarrow \mathbf{X}_0$ ▷ Under attack $\mathbf{X}_0$ remains unchanged
end if

6 NUMERICAL EXPERIMENTATION AND RESULTS

6.1 Experimental setup

We built a testbed, shown in Figure 2, with virtual machines for clients and servers, and a bare-metal switch that uses P4 language[1] to implement L4S AQM, routing, and forwarding functions. This allows us to monitor flows at a packet-level granularity using In-band Network Telemetry (INT). The router runs the P4 implementation of DualPI2, as specified by the IETF. Users can choose what data to collect, and the INT system sends reports to a collector. More details on the testbed can be found in this article [21].

Our testbed; depicted in the Figure 2, has several legitimate clients and servers, some using low-latency services and classic flows. We measure everything from the central L4S-capable router.

[1]P4.org

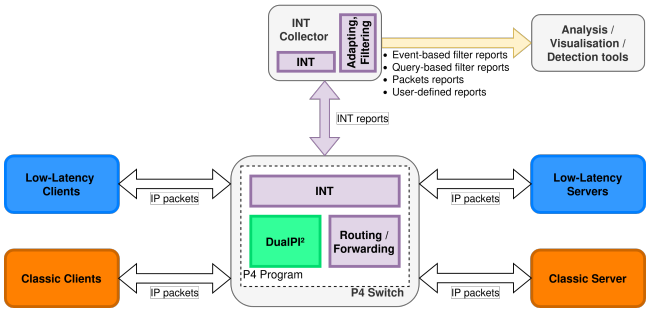

Figure 2: Illustration of our experimental testbed ; note that the number of legitimate users can vary as well as the traffic and the available achievable transmission rate of the router.

We vary the number of legitimate clients and the attacker/regular ratio from 1:1 to 1:10. We also vary the attack power from the minimal attack power, behaving like a legitimate flow by reducing its rate according to the Prague requirements, to the maximum attack power, by not reducing, at all, its rate regardless of the number of ECN signals. This means we can create a range of attack scenarios, from weak attacks with one attack flow hidden in ten legitimate flows, to strong attacks where several attackers compete with one legitimate flow for bandwidth. This setting lets us test different attack types, from subtle to obvious, and see how they affect the network."

We conducted about 200 experiments, each lasting 90 seconds. The experiment starts with 30 seconds of normal traffic, followed by 30 seconds of attack traffic, and ends with 30 seconds of normal traffic again. This design allows us to capture the attack's early stages when its effects are not yet fully visible, making it harder to detect. The return to normal traffic also helps us study the aftermath of an attack, including the time it takes for the router to recover. We collected traffic data during these transition periods, which can be slightly different. We note that this experimental approach generated a lot more normal traffic data than attack data. Some experiments had no attack at all, serving as a reference point. Our large-scale experiment resulted in a total of 714, 967 samples of attack traffic and 3, 901, 982 samples of normal traffic. This means that about 15% of the samples were under attack, while about 85% were normal traffic, as reported in our previous work [21].

6.2 Experimental Detection Results

The very first interesting results concern the relevance of the proposed methodology. We especially want to highlight the importance of the two windows procedure where the first window is used for parameter estimation and the second window is the one over which the detection is carried out. To this end the Figure 3 contrasts the detection results for one single experimentation. Note that the results focus on a small period before and after the attack starts. The red curve shows the detection statistics $\Lambda^\star$ as given in the Equation (15) when not subtracting the mean Log-LR $\lambda_0^\star$ over the estimation window. Clearly, the impact of the attack is easily detectable. Note, however, the non-zero mean before the kick-off of the attack. The average value of the Log-LR changes for every

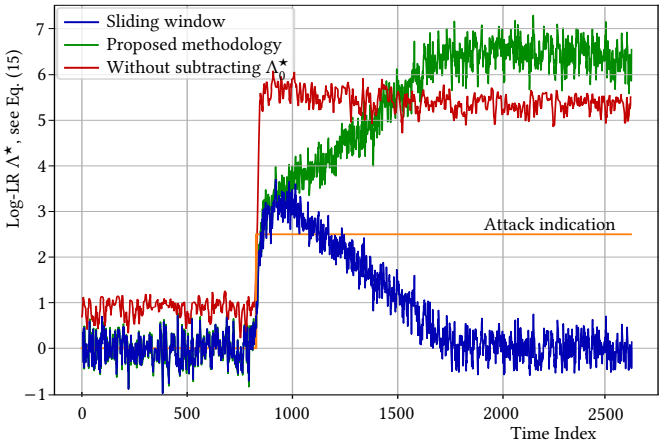

Figure 3: Comparison of the proposed method with ablated versions ; the green curve shows the log-LR $\Lambda_0^s tar$ computed using the proposed methodology ; in red, the methodology is always applied using the last observation for the estimation window even under detection of the attack ; in blue, the proposed methodology is used without subtracting the mean value of the log-LR $\mathbf{a}_0^\top$ estimated over the estimation window.

experimentation, which makes it very difficult to set a threshold τ above which it is assumed that an attack is happening. Indeed, for some experimentation, the mean of the Log-LR $\Lambda^\star$ is large before the attack, creating many false positives. On the opposite, for some other experiments, the mean of the log-LR is very small and the impact of the attack to small preventing the detection of the attack. The blue curve shows the values of the log-LR $\Lambda^\star$ when the estimation window keeps moving even when an attack is detected. Obviously, the attack is clearly detected but its impact fades away very quickly because the estimation windows start containing observations after the attack star. Therefore the adaptive model quickly takes into account the new observations and the impact of the attack is slowly incorporated into the linear model $\mathbf{H}_t$. One can note that is such a case if the attack is not detected as soon as it starts, it is very unlikely to detect it after a small delay.
The green curve shows the value of the proposed overall methodology for calculating the log-LR $\Lambda^\star$ taking into account the average value of the decision statistics over the estimation window and the keeping the same estimation window when an attack is detected. Obviously, the proposed methodology, using the estimation windows to compute the log-LR $\lambda_0^\star$ as in Eq. (16) and as described in Algorithm 1, is very relevant to obtain a standard decision statistic that is almost always centered around zero under legitimate traffic only and preserves the detection possibility even after a few seconds of the attack kick-off.

The most important results about the detection performance of the proposed original methodology are presented in Figure 4. This figure presents the detection performance, using three different window sizes, as ROC (Receiver Operating Characteristics) curves. These curves present the detection accuracy, measured as the true positive rate, also referred to as the power function, the test sensitivity or the recall, as a function of the false alarm rate. Note that for

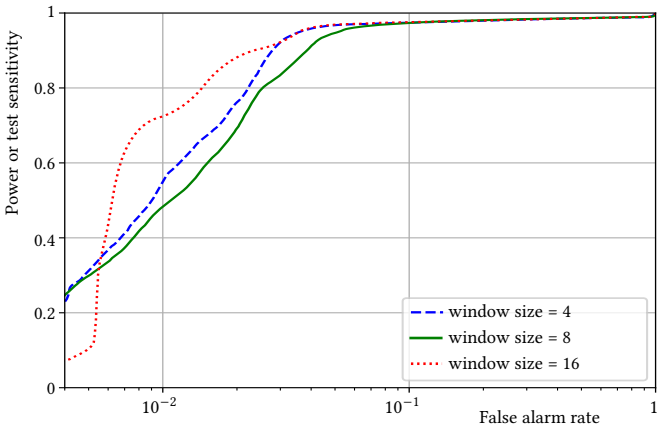

Figure 4: ROC Curves of the proposed detector methodology applied with different window sizes.

readability, the Figure 4 uses a semi-logarithmic scale for the false alarm rate ; this allows better emphasizing the very high detection power achieved for a false alarm rate as small as 0.03 which would be clearly not very visible using a linear scale.
Interestingly, the Figure 4 shows that for a vast majority of the cases the attack on low-latency services can be detected. It also shows that there are small, but not negligible, fraction of cases in which detection is not significantly better as random guessing. This seems to point out that about 3% of network traffic observations under attack are similar to those without attack ; this often occurs at the end of the attack and is in part explainable by issues regarding the labelisation of the data, as sometimes the attack does not stop exactly after 30 seconds.

To show the limit of the proposed methodology, the Figure 5 shows the false alarm rate as a function of the detection threshold τ. Note that, interestingly, Figure 5 shows, in cyan, a comparison with the theoretical false alarm rate one would obtain if the observation

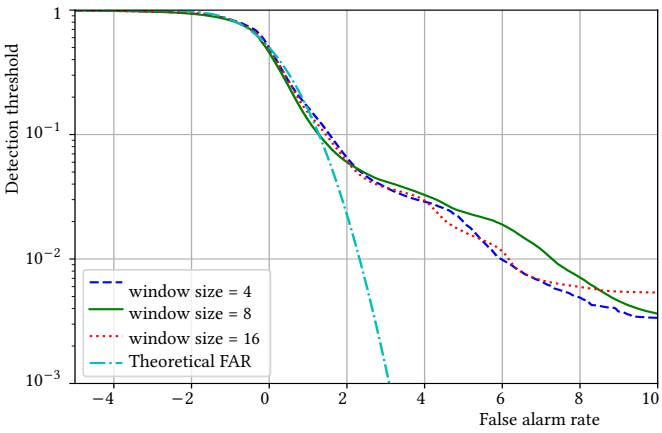

Figure 5: ROC Curves of the proposed simple model with ablation versions. Note that the x-axis representing the false positive rate is plotted using a logarithmic scale.

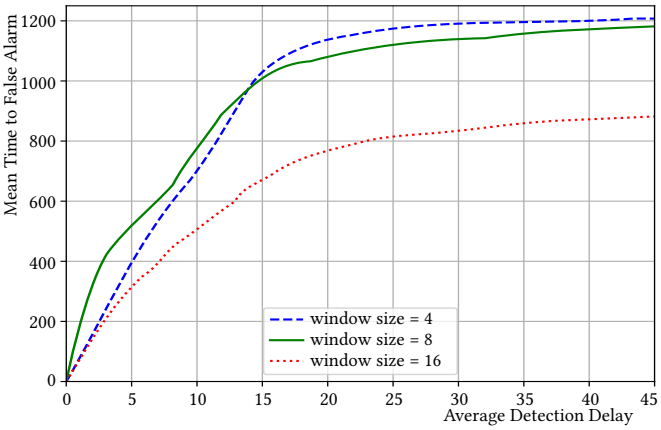

Figure 6: ROC Curves of the proposed simple model with ablation versions. Note that the x-axis representing the false positive rate is plotted using a logarithmic scale.

follows exactly the Gaussian distribution model presented in the Equation (11)–(12). This result clearly emphasizes the limit of the proposed methodology, as the theoretical false alarm rate differs significantly from observations for a false alarm rate smaller than 10%. This shows that the assumed statistical linear model is not accurate enough to allow warranting a false alarm rate setting the decision threshold τ according to the theoretical model. This is due to, in part, to the fact that the observation does not always follow Gaussian distribution and, in part, to the limit of the estimated autoadaptive linear model $\mathbf{H}_t$ to represent the behaviour of the network traffic metrics.

The last interesting results we wanted to show concern the detection delay. Even though the present paper does not propose a specific sequential detection method, such as the well-known CUSUM (cumulative sum), it is interesting to measure, for a given detection threshold τ, the average time before a false alarm is raised and the average delay for detecting the beginning of the attack on low-latency services.

To this end, the Figure 6 shows the result on the detection delay as follows: the average time to false alarm is plotted as a function of the average detection delay. Note that the delays, here, are represented in terms of the number of samples. Ideally, one wishes to minimize the detection delay while having a very large average time to false alarm. Therefore, best results are obtained on the top left corner. Figure 6 shows that the detection delay is relatively small as compared to the average time to false alarm. Similarly, this figure shows that the best results are obtained for a window size of about 8 measurements of the metrics ; a smaller window reduces the detection delay but increases the number of false alarm hence the lower average time to false alarm while, on the opposite, a higher window size delays the detection of the attack.

Note, however, that with the experimentation conducted for this paper is it difficult to measure the average time to false alarm higher than 30 seconds, while in real life conditions such an average period between false alarms is far too small. Also, note that better results would have been obtained using a proper sequential detection method, which is not the scope of the present paper.

7 CONCLUSION AND FUTURE WORKS

The present paper addresses the problem of the detection of attacks on low-latency services in computer networks. This problem is difficult because the legitimate traffic is very complex to model accurately in a general manner. To cope with this issue, we proposed a simple yet efficient autoadaptive linear model for representing the legitimate traffic rather accurately. Using this linear model of the network traffic metrics, we proposed a detection methodology using two sliding windows, the first being used for estimating the parameters of the autoadaptive model and the second is the one over which the presence of the attack is detected.

Over a large set of real-life experimentation, we show that the proposed original methodology, despite its simplicity, achieves rather very good detection performance. However, we have also shown the limitations of the proposed approach for guaranteeing a very false alarm rate. Our next works will focus on using deep learning models in order to compare the performance with the proposed approach as well as to design a sequential detection method to address the problem of the quickest detection under a constraint on the average time to alarm.

ACKNOWLEDGMENTS

This work was partially supported by the French ANR MOSAICO project No. ANR-19-CE25-0012.

REFERENCES

[1] Olga Albisser, Koen De Schepper, Bob Briscoe, Olivier Tilmans, and Henrik Steen. 2019. DUALPI2-Low Latency, Low Loss and Scalable Throughput (L4S) AQM. *Proc. of Netdev* (2019).

[2] Ahmad Azab, Mahmoud Khasawneh, Saed Alrabaee, Kim-Kwang Raymond Choo, and Maysa Sarsour. 2024. Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks* 10, 3 (2024), 676–692.

[3] Sabyasachi Basu, Amarnath Mukherjee, and Steve Klivansky. 1996. Time series models for internet traffic. In *Proceedings of IEEE INFOCOM'96. Conference on Computer Communications*, Vol. 2. IEEE, 611–620.

[4] Dejene BoruOljira, Karl-Johan Grinnemo, Anna Brunstrom, and Javid Taheri. 2020. Validating the sharing behavior and latency characteristics of the L4S architecture. *ACM SIGCOMM Computer Communication Review* 50, 2 (2020), 37–44.

[5] Rémi Cogranne, Guillaume Doyen, Nisrine Ghadban, and Badis Hammi. 2018. Detecting Botclouds at Large Scale: A Decentralized and Robust Detection Method for Multi-Tenant Virtualized Environments. *IEEE Transactions on Network and Service Management* 15, 1 (2018), 68–82. https://doi.org/10.1109/TNSM.2017.2785628

[6] Rémi Cogranne, Quentin Giboulot, and Patrick Bas. 2020. Steganography by Minimizing Statistical Detectability: The cases of JPEG and Color Images. In *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security* (Denver, CO, USA) *(IH&MMSec '20)*. Association for Computing Machinery, New York, NY, USA, 161–167. https://doi.org/10.1145/3369412.3395075

[7] Rémi Cogranne, Quentin Giboulot, and Patrick Bas. 2021. Efficient steganography in JPEG images by minimizing performance of optimal detector. *IEEE Transactions on Information Forensics and Security* 17 (2021), 1328–1343.

[8] Rémi Cogranne and Florent Retraint. 2014. Statistical detection of defects in radiographic images using an adaptive parametric model. *Signal Processing* 96 (2014), 173–189. https://doi.org/10.1016/j.sigpro.2013.09.016

[9] Rémi Cogranne and Florent Retraint. 2014. Statistical detection of defects in radiographic images using an adaptive parametric model. *Signal Processing* 96 (2014), 173–189.

[10] K De Schepper, M Bagnulo, and G White. 2023. RFC 9330: Low Latency, Low Loss, and Scalable Throughput (L4S) Internet Service: Architecture.

[11] Koen De Schepper, Olga Bondarenko, Ing-Jyh Tsang, and Bob Briscoe. 2016. PI2: A Linearized AQM for both Classic and Scalable TCP. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*

(Irvine, California, USA) *(CoNEXT '16)*. Association for Computing Machinery, New York, NY, USA, 105–119. https://doi.org/10.1145/2999572.2999578

[12] Mitra Fouladirad and Igor Nikiforov. 2005. Optimal statistical fault detection with nuisance parameters. *Automatica* 41, 7 (2005), 1157–1171.

[13] Paul M Frank. 1990. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica* 26, 3 (1990), 459–474.

[14] Alice Hutchings and Richard Clayton. 2016. Exploring the provision of online booter services. *Deviant Behavior* 37, 10 (2016), 1163–1178.

[15] E.L. Lehmann and J.P. Romano. 2005. *Testing Statistical Hypotheses, Second Edition* (3rd ed.). Springer.

[16] Marius Letourneau, Guillaume Doyen, Rémi Cogranne, and Bertrand Mathieu. 2023. A comprehensive characterization of threats targeting low-latency services: the case of L4S. *Journal of Network and Systems Management* 31, 1 (2023), 19.

[17] Marius Letourneau, Kouame Boris N'Djore, Guillaume Doyen, Bertrand Mathieu, Rémi Cogranne, and Huu Nghia Nguyen. 2021. Assessing the threats targeting low latency traffic: the case of L4S. In *2021 17th International Conference on Network and Service Management (CNSM)*. IEEE, 544–550.

[18] Lian Lian. 2024. Network traffic prediction model based on linear and nonlinear model combination. *ETRI Journal* 46, 3 (2024), 461–472. https://doi.org/10.4218/etrij.2023-0136 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.4218/etrij.2023-0136

[19] Bertrand Mathieu and Stéphane Tuffin. 2021. Evaluating the L4S Architecture in Cellular Networks with a Programmable Switch. In *2021 IEEE Symposium on Computers and Communications (ISCC)*. 1–6. https://doi.org/10.1109/ISCC53001.2021.9631539

[20] Szilveszter Nádas, Gergő Gombos, Ferenc Fejes, and Sándor Laki. 2020. A Congestion Control Independent L4S Scheduler. In *Proceedings of the 2020 Applied Networking Research Workshop* (Virtual Event, Spain) *(ANRW '20)*. Association for Computing Machinery, New York, NY, USA, 45–51. https://doi.org/10.1145/3404868.3406669

[21] Huu Nghia Nguyen, Bertrand Mathieu, Marius Letourneau, Guillaume Doyen, Stéphane Tuffin, and Edgardo Montes de Oca. 2023. A Comprehensive P4-based Monitoring Framework for L4S leveraging In-band Network Telemetry. In *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 1–6.

[22] Tan Nguyen, Remi Cogranne, and Guillaume Doyen. 2015. An optimal statistical test for robust detection against interest flooding attacks in ccn. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 252–260.

[23] Tan Nguyen, Hoang-Long Mai, Rémi Cogranne, Guillaume Doyen, Wissam Mallouli, Luong Nguyen, Moustapha El Aoun, Edgardo Montes De Oca, and Olivier Festor. 2019. Reliable Detection of Interest Flooding Attack in Real Deployment of Named Data Networking. *IEEE Transactions on Information Forensics and Security* 14, 9 (Sept. 2019), 2470–2489. https://doi.org/10.1109/TIFS.2019.2899247

[24] Tan Nguyen, Hoang-Long Mai, Guillaume Doyen, Rémi Cogranne, Wissam Mallouli, Edgardo Montes De Oca, and Olivier Festor. 2018. A security monitoring plane for named data networking deployment. *IEEE Communications Magazine* 56, 11 (2018), 88–94.

[25] Fatih Berkay Sarpkaya, Ashutosh Srivastava, Fraida Fund, and Shivendra Panwar. 2024. To switch or not to switch to TCP Prague? Incentives for adoption in a partial L4S deployment. In *Proceedings of the 2024 Applied Networking Research Workshop* (Vancouver, AA, Canada) *(ANRW '24)*. Association for Computing Machinery, New York, NY, USA, 45–52. https://doi.org/10.1145/3673422.3674896

[26] L.L. Scharf and B. Friedlander. 1994. Matched subspace detectors. *IEEE Transactions on Signal Processing* 42, 8 (1994), 2146–2157. https://doi.org/10.1109/78.301849

[27] K. De Schepper, B. Briscoe, and G. White. 2009. DualQ Coupled AQMs for Low Latency, Low Loss and Scalable Throughput (L4S) Internet Service: Architecture.

[28] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. 2015. Content-adaptive steganography by minimizing statistical detectability. *IEEE Transactions on Information Forensics and Security* 11, 2 (2015), 221–234.

[29] Vahid Sedighi, Jessica Fridrich, and Rémi Cogranne. 2015. Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model. In *Media Watermarking, Security, and Forensics 2015*, Vol. 9409. SPIE, 144–156.

[30] Karim Tout, Rémi Cogranne, and Florent Retraint. 2018. Statistical decision methods in the presence of linear nuisance parameters and despite imaging system heteroscedastic noise: Application to wheel surface inspection. *Signal Processing* 144 (2018), 430–443. https://doi.org/10.1016/j.sigpro.2017.10.030

[31] Hao Yin, Chuang Lin, Berton Sebastien, Bo Li, and Geyong Min. 2005. Network traffic prediction based on a new time series model. *International Journal of Communication Systems* 18, 8 (2005), 711–729.

submitted 05 Novembre 2024; revised 16 Decembre 2024